

Distributed Operating Systems

→ - 1) It does not exist

1980s } - 2) It does ... but its in
1990s } regular operating systems.

(Middleware)

→ message passing

→ remote procedure calls

→ shared resources

→ client-server architectures
(and so on)

“Distributed”

- No shared memory
- multiple computing units
- network

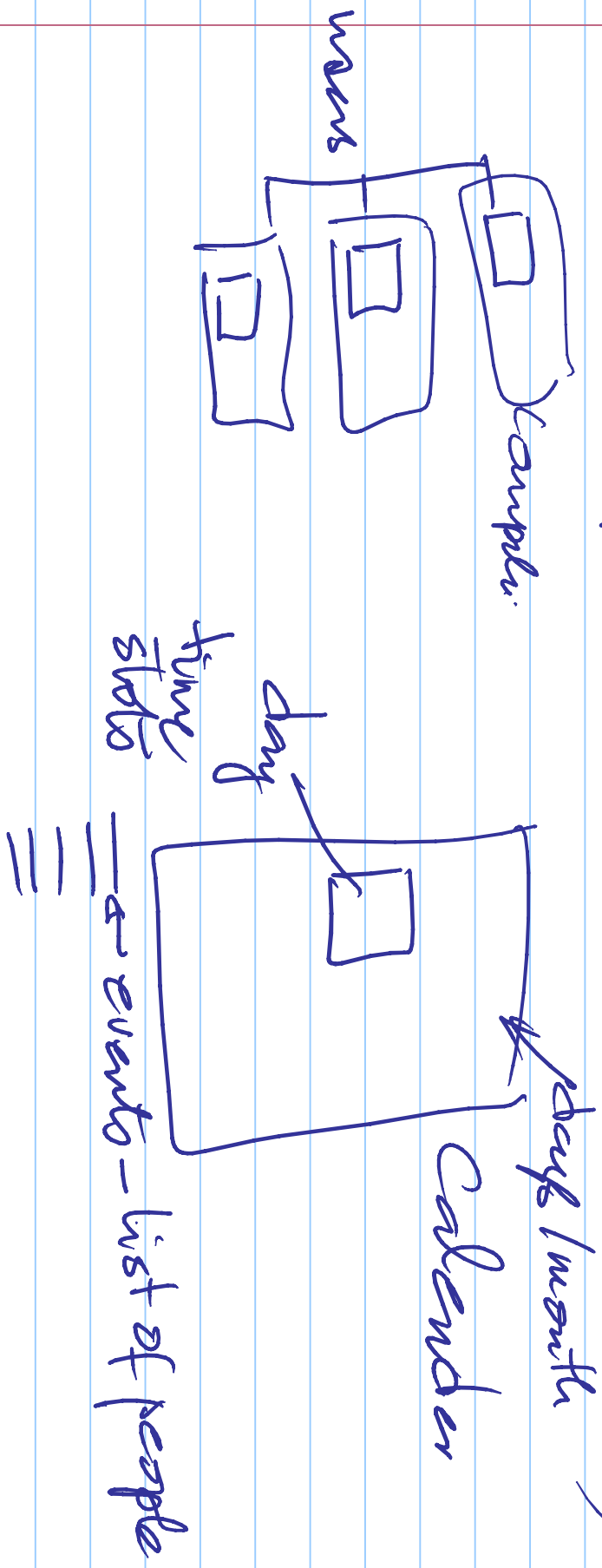
Andy Tannenbaum

A distributed OS is a collection of

↳ autonomous components that appear
to the user as one integrated
system.

↳ Separate
OS.

Calendar application (distributed)



Everybody
has a copy
of Calendar

↙ replicated

Everyone has
a copy of all
events he/she
is involved in

↙ partitioned

Calendar
stored &
updated on
one designated
component

↙ distributed
↙ centralized

Replicated → lookups are fast

→ updates? atomically?

Partitioned → updates are easy / easier
→ lookups are hard / slow

✓ Single point of failure
Can be realized - all operations are remote
but no inconsistencies

Distributed OS models

- Workstation model
 - means we workstations, the system is the collection of workstation
- Server Model → all computation & data located at a server
user computers are "thin clients"
- Internet Cloud Model

Operating System

- local kernel (CPU, mem, I/O)

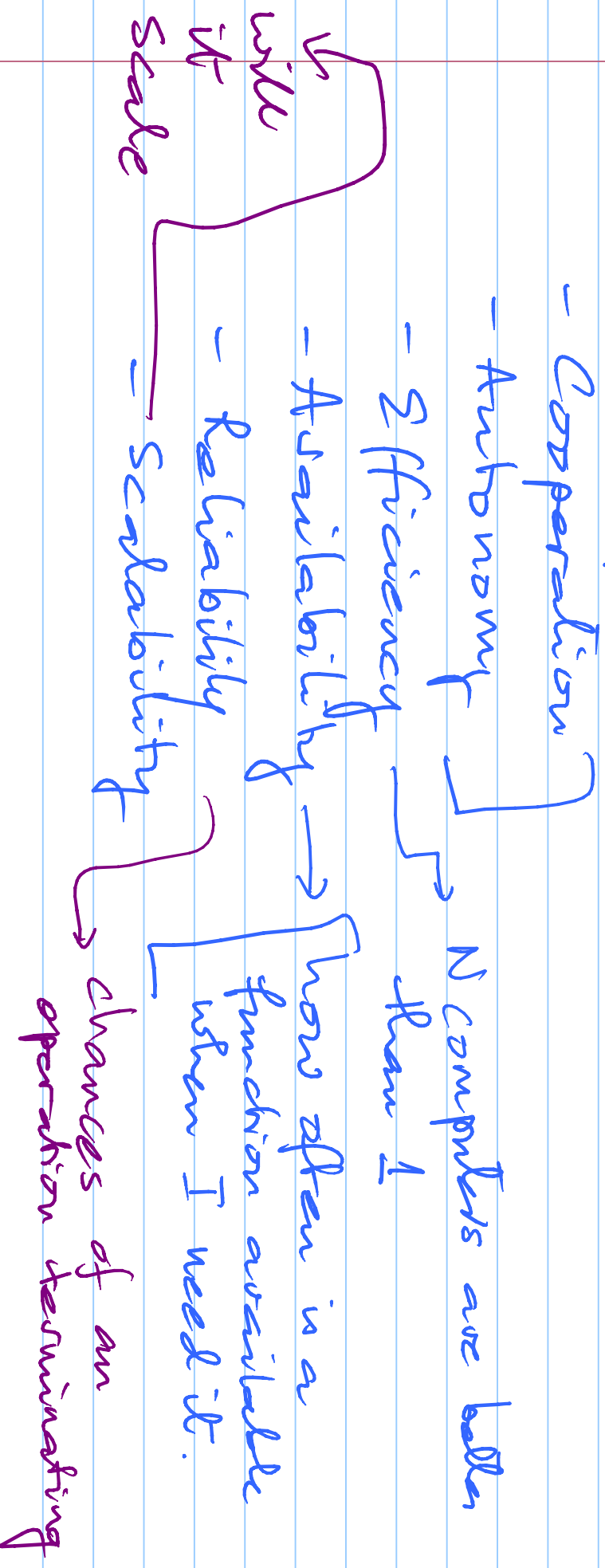
+ networking → messaging

→ RPC

→ Dist Shared Memory

+ sharing (file systems, remote execution & so on)

Properties of Distribution



Issues

- Granularity → fine — bad
→ coarse → good
- Dist Application design
 - eg calendar
- Dist Databases →
- Replication, partitioning, distributed

- Reduce Server computation
 - client side scripting
- Reduce communication
 - client caching
- Latency (Network)
 - ? - high speed often means high latency